Verfahren zur Durchführung eines Software-Updates eines elektronischen Steuergerätes durch eine Flash-Programmierung über eine serielle Schnittstelle und ein entsprechender Zustandsautomat

Die vorliegende Erfindung betrifft ein Verfahren zur Durchführung eines Software-Updates eines elektronischen Steuergerätes durch eine Flash-Programmierung über eine serielle Schnittstelle.

Stand der Technik

Der Einsatz eines sogenannten Flash als Speichertechnologie für Programm- und Datenstand nimmt in elektronischen Steuergeräten zu. Diese Speichertechnologie ermöglicht ein Software-Update der Steuergeräte durch eine Neuprogrammierung des entsprechenden Flash-Speichers der Steuergeräte über serielle Schnittstellen. Bei der seriellen Schnittstelle kann es sich dabei bspw. um eine zentrale Off-Board-Diagnoseschnittstelle eines Fahrzeugs handeln, über welche mit einem sog. Flash-Programmierwerkzeug der Flash-Speicher eines elektronischen

Steuergerätes des Fahrzeuges neu programmiert wird. Somit ist ein Software-Update ohne Ausbau des entsprechenden elektronischen Steuergerätes aus dem Fahrzeug möglich, was zu erheblichen Kosteneinsparungen gegenüber einem Steuergeräteaustausch bzw. -ausbaus führt. Bei der beschriebenen Art der Flash-Programmierung sind insbesondere im Service der Fahrzeuge sowie im Bereich sicherheitsrelevanter elektronischer Steuergeräte hohe Sicherheits- und Zuverlässigkeitsanforderungen zu erfüllen.

Mit den derzeit eingesetzten Flash-Technologien können nur ganze Flash-Bereiche eines Flash-Speichers gelöscht oder neu programmiert werden. Dabei wird eine kleinste, physikalisch zusammengehörende, geschlossen lösch- oder programmierbare Speichereinheit des Flash-Speichers als Segment bezeichnet. Bei einer Flash-Programmierung sind deshalb die Schritte Löschen und Programmieren von Flash-Segmenten zu unterscheiden. Dabei muß ferner beachtet werden, daß es nicht möglich ist gleichzeitig aus einem Flash-Segment ein Programm auszuführen, während ein anderes Flash-Segment des gleichen Flash-Bausteins neu programmiert wird. Die Programmteile zur Steuerung des Programmierablaufs für ein Flash-Bauteil müssen deshalb, zumindest temporär während der Durchführung der Flash-Programmierung, in einen anderen Speicherbaustein, bspw. in einen anderen Flash-Baustein oder einen freien RAM (Random Access Memory)-Bereich des Steuergerätes ausgelagert werden.

Wegen der begrenzten Übertragungsleistung der Off-Board-Diagnoseschnittstelle kommt es bei großen Flash-Speichern von elektronischen Steuergeräten zu recht langen Flash-Programmierzeiten. Deshalb besteht in der Produktion und im

Service häufig die Anforderung, die Flash-Programmierzeiten zu verkürzen.

Ferner ist bei einer Flash-Programmierung aus Haftungsgründen stets zu beachten, daß eine nicht autorisierte Flash-Programmierung oder eine Flash-Programmierung mit einem manipulierten Programm- oder Datenstand möglichst zu verhindern ist. Letztlich ist zu beachten, daß eine Flash-Programmierung über eine genannte Off-Board-Diagnoseschnittstelle stets ein verhältnismäßig lange Zeitspanne in Anspruch nehmen kann. Dabei ist mit Abbrüchen des Programmierablaufs durch evtl. auftretende Störungen jederzeit zu rechnen. Derartige Störungen sind etwa der Ausfall der Spannungsversorgung eines Fahrzeuges oder des Flash-Programmierwerkzeuges, unzulässige Reaktion anderer Steuergeräte im Netzwerk, Unterbrechung der Kommunikationsverbindung zwischen dem zu programmierenden elektronischen Steuergerät und dem dazu eingesetzten Flash-Programmierwerkzeug oder ein Bedienfehler. Auch eine fehlgeschlagende Authentisierung und Signaturprüfung können zum Abbruch einer Flash-Programmierung führen. Deshalb ist es nötig die Verfügbarkeit bzw. einen sofortigen Neustart der Flash-Programmierung jederzeit gewähren zu können.

Vorteile der Erfindung

Es wird ein erfindungsgemäßes Verfahren gemäß Anspruch 1 und ein entsprechender Zustandsautomat gemäß Anspruch 8 vorgestellt. Weitere Vorteile und bevorzugte Ausführungsformen werden in den entsprechenden Unteransprüchen aufgeführt.

Gemäß Anspruch 1 wird ein Verfahren zur Durchführung eines Software-Updates eines Steuergerätes durch eine Flash-

Programmierung eines mehrere Segmente aufweisenden Flashspeichers des Steuergerätes über eine serielle Schnittstelle bereitgestellt, wobei in einem ersten Schritt des Verfahrens an die Flash-Programmierung zu stellende Anforderungen festgelegt werden, so daß ein Ablauf der Flash-Programmierung durch einen Zustände und Übergänge der Software des Steuergerätes definierenden Zustandsautomaten spezifiziert und letztlich Verfügbarkeits-, Sicherheits- und Zuverlässigkeitsanforderungen eines jeden Zustandes und eines jeden Übergangs des Zustandsautomatischen überprüft werden.

Vorzugsweise werden bei Festlegen von an die Flash-Programmierung zu stellende Anforderungen zunächst für die Software des Steuergerätes verschiedene Betriebszustände spezifiziert. Dabei wird vorzugsweise unterschieden zwischen einem "Anfangszustand", einem "Normalzustand" und einem Zustand "Software-Update". Ferner werden die Übergänge zwischen den genannten Betriebszuständen und die Übergangsbedingungen definiert. Bei einer weiteren bevorzugten Ausführungsform des Verfahrens werden für die Flash-Programmierung relevante Speicherblöcke der Software des Steuergerätes in programmierbare und nicht programmierbare Speicherblöcke unterteilt und neu zu programmierende Komponenten der Software den Steuerblöcken entsprechend zugeordnet. Weiterhin vorzugsweise werden die Speicherblöcke der Software jeweils einem Speicher des Steuergerätes, insbesondere ein programmierbarer Speicherblock einem Segment des Flash-Speichers bzw. ein nicht programmierbarer Speicherblock einem ROM (Read-Only-Memory) des elektronischen Steuergerätes zugeordnet. Aufgrund der begrenzten Übertragungsleistung der Off-Board-Diagnoseschnittstelle kommt es bei großen Flash-Speichern zu recht langen Flash-Programmierzeiten. Deshalb ist es

wünschenswert, die Flash-Programmierzeiten zu verkürzen, was bspw. durch eine Verringerung der neu zu programmierenden Flash-Segmente möglich ist. Dies wird vorzugsweise durch die Flash-Programmierung einzelner Software-Funktionen oder durch eine getrennte Flash-Programmierung für den Programm- und Datenstand des elektronischen Steuergerätes erreicht. Dabei wird häufig der Programmstand bereits bei der Steuergeräteproduktion programmiert, während der Datenstand später bspw. fahrzeugspezifisch am Ende der Produktion eines Fahrzeuges programmiert wird. Aufgrund dessen werden in einer weiteren bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens jeweils in Segmenten des Flash-Speichers des Steuergerätes der sog. Boot-Block, der Programm- und er Datenstand abgelegt. Das bedeutet, daß verschiedene Softwarefunktionen, sowie Programm- und Datenstand in verschiedenen Flash-Segmenten abgelegt werden. Alle Programmteile des Steuergerätes, die für eine Kommunikation zwischen dem Steuergerät und einem Flash-Programmierwerkzeug über die Off-Board-Diagnoseschnittstelle während einer Flash-Programmierung erforderlich sind, müssen dabei zusammen mit entsprechenden Flash-Programmierroutinen, einem sog. Flash-Loader im ROM des elektronischen Steuergeräts oder in einem anderen weiteren Flash-Segment abgelegt werden. Die für die Kommunikation zwischen Steuergerät und Flash-Programmierwerkzeug erforderlichen Programmteile werden unterteilt in programmierbare und nicht programmierbare Anteile, nämlich einen im ROM abgelegten Basisumfang im folgenden als Start-up-Block bezeichnet, und einen im Flash abgelegten Basisumfang, im folgenden als Boot-Block bezeichnet. Start-up- und Boot-Block zusammen stellen die für eine Flash-Programmierung über eine Off-Board-Diagnoseschnittstelle notwendige Software-Funktionalität

eines Mikrocontrollers des Steuergerätes zur Verfügung. Eine Aufteilung in Start-up- und Boot-Block ist aus verschiedenen Gründen sinnvoll. So kann der Boot-Block selbst, falls er, wie beschrieben, im Flash-Speicher abgelegt wird, neu programmiert werden. Ferner kann im Boot-Block, der aktuelle Status einer Flash-Programmierung unverlierbar abgespeichert werden, so daß bspw. nach einem Abbruch der Flash-Programmierung ein Wiederaufsetzen möglich ist. Die unveränderbare Basisfunktionalität des Start-up-Blocks und eine Kennung für eine Hardwarevariante des elektronischen Steuergerätes können hingegen im kostengünstigeren und nicht neu programmierbaren ROM des Steuergerätes abgelegt werden. Erfindungsgemäß wird ferner der Programm- und der Datenstand jeweils in einem anderen Segment des Flash-Speichers abgelegt.

In einer weiteren bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens werden Sicherheits-, Zuverlässigkeits- und Verfügbarkeitsanforderungen der durchzuführenden Flash-Programmierung spezifiziert. Ein Übergang eines Mikrocontrollers des Steuergerätes in den Betriebszustand "Software-Update" wird von einem Flash-Programmierwerkzeug angestoßen. Neben evtl. notwendig Plausibilitätsprüfungen, wie etwa bei Motorsteuergeräten die Prüfung auf einen Motorstillstand, die vor einem Beenden eines Fahrprogramms und einem Übergang in den Betriebszustand "Software-Update" durchgeführt werden müssen, sind bei einem Einsatz in Produktion und im Service weitere Sicherheitsmaßnahmen erforderlich. Demnach ist es bspw. aus Haftungsgründen erforderlich, eine nicht autorisierte Flash-Programmierung oder Flash-Programmierung mit manipuliertem Programm- oder Datenstand möglichst zu verhindern. Zumindest sollten derartige Flash-Programmierungen erkannt und nachgewiesen werden können.

Daher wird ein Flash-Programmierzugriff in der Regel über zwei unterschiedliche Verschlüsselungsverfahren abgesichert. Zum Einen handelt es sich dabei um eine Authentisierung, was einer Prüfung der eigentlichen Zugriffsberechtigung entspricht und nach einer Plausibilitätsprüfung durchgeführt wird. Dabei wird anhand eines digitalen Schlüssels überprüft, ob ein Anwender des Flash-Programmierwerkzeuges überhaupt berechtigt ist, ein Software-Update durchzuführen. Ein zweites Verschlüsselungsverfahren ist eine sog. Signaturprüfung. Hierbei wird die Datenkonsistenz eines neu zu programmierenden Programm- oder Datenstands überprüft.

Bei der Signaturprüfung wird von einem Flash-Programmierwerkzeug anhand eines weiteren digitalen Schlüssels überprüft, ob der neu zu programmierende Programm- oder Datenstand zur Steuergeräte-Hardware paßt und ob der neu zu programmierende Programm- oder Datenstand bspw. nach der Auslieferung durch den Fahrzeughersteller an die Service-Organisation unzulässig manipuliert wurde. Erst nach einem erfolgreichen Abschluß bei der genannten Prüfung soll das eigentliche Löschen und Programmieren der entsprechenden Segmente des Flash-Speichers ermöglicht bzw. freigegeben werden. Die Freigabe erfolgt dabei durch den vorstehend beschriebenen Boot-Block. Bei der Spezifizierung des Sicherheits- und Zuverlässigkeitsanforderung der Flash-Programmierung ist auch zu beachten, daß nach der Flash-Programmierung die Signatur eines Mikrocontrollers des Steuergerätes auf Basis des tatsächlich in den Flash-Speicher programmierten Programm- und Datenstands berechnet wird, um Fehler während der Programmierung erkennen zu können. Nach einer erfolgreichen Signaturprüfung wird diese berechnete Signatur selbst im Flash-Speicher abgelegt. Dazu werden besondere Speicherstrukturen, eine sog.

Programmstands- und Datenstandslogistik als Teil des Programm- und des Datenstands im Flash-Speicher abgelegt. Nur nach einer erfolgreichen Signaturprüfung gibt der Boot-Block die Aktivierung des neuen Programms wie bspw. eines Fahrprogramms frei.

Ferner wird bei dem erfindungsgemäßen Verfahren vorzugsweise auch die Verfügbarkeitsanforderung der Flash-Programmierung spezifiziert. Da die Flash-Programmierung über die Off-Board-Diagnoseschnittstelle trotz bereits beschriebener Optimierungsmaßnahmen eine verhältnismäßig lange Zeitspanne in Anspruch nehmen kann, ist generell mit Abbrüchen des Programmierablaufs durch Störungen jederzeit zu rechnen. Derartige Störungen sind etwa ein Ausfall einer Spannungsversorgung eines Fahrzeuges oder eines Flash-Programmierwerkzeugs, unzulässige Reaktionen anderer Störgeräte im Netzwerk, Unterbrechungen der Kommunikationsverbindung zwischen dem elektronischen Steuergerät und dem eingesetzten Flash-Programmierwerkzeug oder Bedienfehler. Auch fehlgeschlagene Authentisierung und Signaturprüfungen führen in der Regel zu einem Abbruch der Flash-Programmierung. Für einen Entwurf des Ablaufs der Flash-Programmierung ist es deshalb wichtig, die Verfügbarkeit der Flash-Programmierung unter allen denkbaren Umständen zu gewährleisten. Dies bedeutet bspw., daß nach einem Abbruch in allen Situationen jederzeit ein Neustart des Programmierablaufs gewährleistet wird. Dazu wird in einer weiteren bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens durch den Zustandsautomaten bei Durchführung der Flash-Programmierung im Betriebszustand "Software-Update" einnehmbare Subzustände, Übergänge zwischen diesen und Übergangsbedingungen spezifiziert. Als Subzustände kann es sich dabei um den Subzustand "Abbruch/Fehlermeldung" oder

"Abschluß/Erfolgsmeldung" handeln. Ferner können vorzugsweise Subzustände für Authentisierung und Signaturprüfung spezifiziert werden sowie Subzustände für das Löschen und Programmieren von Segmenten des Flash-Speichers. Weiter ist es wünschenswert eine Spezifikation von Subzuständen für eine Auslagerung und eine Flash-Programmierung des Boot-Blocks vorzunehmen. Eine Spezifikation von Übergängen zwischen den genannten Subzuständen und entsprechender Übergangsbedingungen wird erfindungsgemäß ebenfalls vorgenommen.

Ferner umfaß die vorliegende Erfindung ein Computerprogramm bestehend aus Programmcodeelementen, durch welches bei Ausführung der Programmcodeelemente auf einem Computer oder auf einem Computersystem automatisch vordefinierte Verfügbarkeits-, Sicherheits- und Zuverlässigkeitsanforderungen eines jeden Zustands und eines jeden Übergangs eines vorstehend beschriebenen Zustandsautomaten überprüft werden.

Letztlich betrifft die vorliegende Erfindung ein Verfahren zur Flash-Programmierung eines vorstehend beschriebenen Boot-Blocks. Es wird ein Verfahren zur Durchführung einer Flash-Programmierung eines für die Durchführung der Flash-Programmierung notwendige Software-Funktionalität bereitstellenden Boot-Blocks bereitgestellt. Der Boot-Block ist dabei in einem ersten Segment eines Flash-Speichers abgelegt. In einem ersten Schritt wird der neu zu programmierende alte Boot-Block in einen freien RAM-Bereich kopiert. D.h. der noch aktive alte Boot-Block muß während der Flash-Programmierung in einen anderen Speicherbaustein des Steuergerätes ausgelagert werden, was bedeutet daß der Boot-Block relokatierbar sein muß. In einem zweiten Schritt wird sodann der alte Boot-Block im RAM aktiviert und im

Flash-Speicher, wo er in einem ersten Segment abgelegt ist, deaktiviert. Weiterhin wird der neue Boot-Block in einem zweiten Segment des Flash-Speichers zwischenabgelegt. Dieser Schritt umfaßt dabei das Löschen des zweiten Segmentes des Flash-Speichers, Programmieren des neuen Boot-Blocks in das zweite Segment des Flash-Speichers und eine Signaturprüfung für den neuen Boot-Block in dem zweiten Segment des Flash-Speichers. Nach einem Abbruch während dieser Verfahrensschritte kann mit dem gültigen, alten Boot-Block in dem ersten Segment des Flash-Speichers die Flash-Programmierung erneut gestartet werden. In einem weiteren Schritt des erfindungsgemäßen Verfahrens wird letztlich der neue Boot-Block programmiert durch Kopieren des zweiten Segmentes des Flash-Speichers in das erste Segment des Flash-Speichers. Dieser Schritt umfaßt dabei das Löschen des ersten Flash-Segmentes, Programmieren des neuen Boot-Blocks in das erste Flash-Segment durch Kopieren des zweiten Flash-Segmentes in das erste Flash-Segment und eine Signaturprüfung für den neuen Boot-Block in dem ersten Flash-Segment. Nach einem Abbruch während dieser Verfahrensschritte kann mit dem gültigen, neuen Boot-Block in dem zweiten Flash-Segment die Flash-Programmierung erneut gestartet werden. Vorzugsweise wird im Flash-Speicher immer ein Boot-Block als für einen Neustart der Flash-Programmierung gültiger Boot-Block markiert. Diese Gültigkeitsmarkierung selbst muß dabei unverlierbar im Flash-Speicher abgelegt werden, so daß mit dieser Information ein Wiederaufsetzen möglich ist. In einem letzten Schritt des erfindungsgemäßen Verfahrens wird sodann der neue Boot-Block im ersten Segment des Flash-Speichers aktiviert und gleichzeitig der alte Boot-Block im RAM deaktiviert.

Weiter Vorteile und bevorzugte Ausführungsformen der Erfindung werden anhand der folgenden Figuren näher erläutert:

Figur 1 schematische Darstellung einer Spezifikation von für eine Flash-Programmierung relevanten Speicherblöcken eines Steuergerätes gemäß einer Ausführungsform des erfindungsgemäßen Verfahrens;

Figur 2 schematische Darstellung einer Spezifikation von Sicherheitsanforderungen und -maßnahmen gemäß einer weiteren Ausführungsform des erfindungsgemäßen Verfahrens;

Figur 3 schematische Darstellung von Zuständen und Übergängen eines Boot-Blocks bei einer Flash-Programmierung von Programm- und Datenstand eines elektronischen Steuergerätes;

Figur 4 schematische Darstellung des Ablaufs einer Ausführungsform eines erfindungsgemäßen Verfahrens zur Durchführung einer Flash-Programmierung eines Boot-Blocks.

Figur 1 zeigt eine Zuordnung von Speicherblöcken einer Software eines Steuergerätes für eine Durchführung eines Software-Updates eines Steuergerätes durch eine Flash-Programmierung. Gezeigt ist ein Steuergerät 1 mit einem Mikrocontroller 2. Der Mikrocontroller 2 verfügt über einen Mikroprozessor 3 und drei verschiedene Speicher nämlich einen ROM (Read-Only-Memory) 4, einen Flash-Speicher 5 und einen RAM (Random Access Memory) 6. Ferner weist das Steuergerät 1 eine serielle Schnittstelle 7 zur Ankopplung an eine Off-Board-Diagnoseschnittstelle 8 auf, über welche

ein Flash-Programmierwerkzeug angeschlossen werden kann. Im unteren Teil von Figur 1 ist eine Speicherzuteilung von für die Flash-Programmierung relevanten Speicherblöcken der Software des Steuergerätes 1 dargestellt. Dabei werden die Speicherblöcke in programmierbare und nicht programmierbare Speicherblöcke unterteilt und neu zu programmierende Komponenten der Software den Speicherblöcken entsprechend zugeordnet. Programmteile des Mikrocontrollers 2, die für eine Kommunikation zwischen dem Mikrocontroller 2 und einem Flash-Programmierwerkzeug über die Off-Board-Diagnoseschnittstelle 8 während der Flash-Programmierung erforderlich sind, werden unterteilt in einen sog. Start-up-Block 9 und einen sog. Boot-Block 10. Der Start-up-Block 9 und der Boot-Block 10 stellen zusammen die für die Flash-Programmierung über die Off-Board-Diagnoseschnittstelle 8 notwendige Softwarefunktionalität des Mikrocontrollers 2 zur Verfügung. Die Aufteilung in Start-up-Block 9 und Boot-Block 10 ist aus verschiedenen Gründen sinnvoll. So kann der Boot-Block 10 selbst, der im hier dargestellten Fall in einem Segment A des Flash-Speichers 11 abgelegt ist, neu programmiert werden. Außerdem kann im Boot-Block 10 der aktuelle Status der Flash-Programmierung unverlierbar abgespeichert werden, so daß bspw. nach einem Abbruch ein Wiederaufsetzen möglich ist. Die unveränderbare Basisfunktionalität des Start-up-Blocks 9 kann dagegen im kostengünstigeren und nicht neu programmierbaren ROM 12 abgelegt werden. In einem weiteren Segment des Flash-Speichers, einem Flash-Segment B wird der Programmstand abgelegt und in einem Flash-Segment C der Datenstand.

In Figur 2 ist eine Spezifikation von Sicherheitsanforderungen bei Durchführung einer Flash-Programmierung dargestellt. Gezeigt ist ein möglicher Ablauf einer Kommunikation zwischen einem Flash-

Programmierwerkzeug 13 und einem Mikrocontroller 2 eines Steuergerätes. Nach einer durch Anfrage seitens des Flash-Programmierwerkzeugs 13 und Rückmeldung des Mikrocontrollers 2 durchgeführten Plausibilitätsprüfung 14, die vor einem Übergang in den Betriebszustand "Software-Update" durchgeführt werden muß, wird eine Prüfung bzgl. der eigentlichen Zugriffberechtigung durchgeführt. Dieser Schritt wird als Authentisierung 15 bezeichnet. Dabei wird anhand eines digitalen Schlüssels überprüft, ob ein Anwender des Flash-Programmierwerkzeugs 13 berechtigt ist, ein Software-Update vorzunehmen. In einem weiteren Prüfungsschritt 16 wird die Datenkonsistenz des neu zu programmierenden Programm- oder Datenstands überprüft. Dieser Schritt wird auch als Signaturprüfung bezeichnet. Hierbei wird vom Flash-Programmierwerkzeug 13 anhand eines weiteren digitalen Schlüssels überprüft, ob der neu zu programmierende Programm- oder Datenstand zur Steuergerätehardware paßt und ob der neu zu programmierende Programm- oder Datenstand seit seiner Auslieferung unzulässig manipuliert wurde. Erst nach einem erfolgreichen Abschluß bei der Prüfung werden die Flash-Segmente in einem Schritt 17 gelöscht und anschließend in einem Schritt 18 die entsprechenden Flash-Segmente programmiert. Nach der Flash-Programmierung wird die Signatur vom Mikrocontroller 2 auf Basis des tatsächlich im Flash-Speicher programmmierten Programm- und Datenstands berechnet, um Fehler während der Programmierung erkennen zu können. Nach erfolgreicher Signaturprüfung 19 wird diese berechnete Signaturprüfung selbst im Flash-Speicher abgelegt. Dazu werden besondere Speicherstrukturen, sog. Programmstands- und Datenstandslogistik als Teil des Programm- und des Datenstands im Flash-Speicher abgelegt. Nur nach einer erfolgreichen Signaturprüfung 19 gibt der Boot-Block die

Aktivierung des neuen Programms wie bspw. eines Fahrprogramms frei.

Figur 3 zeigt in schematischer Darstellung Zustand und Übergänge eines Boot-Blocks bei einer Flash-Programmierung von Programm- und Datenstand. Zunächst wird in einem Schritt 20 bei Ankopplung eines Flash-Programmierwerkzeugs an den Mikrocontroller über eine Off-Board-Diagnoseschnittstelle das Steuergerät identifiziert und ein Übergang des Mikrocontrollers in dem Betriebszustand "Software-Update" initiiert. Wird hierbei in einem Schritt 21 ein Fehler erkannt, so kommt es sofort zu einem Abbruch des Programmiervorgangs mit gleichzeitiger Ausgabe einer Fehlermeldung F. In einem weiteren Schritt 22 wird eine Authentisierung des Benutzers des angekoppelten Flash-Programmierwerkzeuges vorgenommen. Auch hier kommt es zu einem Abbruch mit einer Fehlermeldung F, falls in einem Schritt 23 ein Fehler erkannt wird. Im Anschluß daran wird eine Signaturprüfung 24 vorgenommen, was einhergeht mit einer Prüfung der Datenkonsistenz über Hardware-/Programmstands-/ Datenstands-Logistik. Ein erkannter Fehler 25 wird auch hier mit einem Abbruch und einhergehender Fehlermeldung F signalisiert. Nach Durchführung dieser Schritte kommt es zu einem Löschen 26 des Flash-Segmentes, in welchem der Programmstand abgelegt ist, daraufhin wird in eine Schritt 27 der neue Programmstand programmiert und eine Signaturprüfung 28 für den neuen Programmstand durchgeführt. Die gleichen Schritte werden in den Schritten 29, 30, 31 bzgl. der Flash-Programmierung des Datenstandes vorgenommen. Wird bei der Signaturprüfung für Programmstand bzw. für Datenstand ein Fehler erkannt, so erfolgt auch hier ein Abbruch mit einer einhergehenden Fehlermeldung F. Werden demgegenüber keine Fehler erkannt, so erfolgt in einem Schritt 32 ein Übergang

des Mikrocontrollers in dem Betriebszustand "Anfangszustand" durch ein Reset.

Figur 4 beschreibt die Verfahrensschritte bei einer Flash-Programmierung eines Boot-Blocks. Zunächst muß der aktive Boot-Block "A" während der Flash-Programmierung in einen anderen Speicherbaustein des Mikrocontrollers ausgelagert werden, d.h. Boot-Block "A" muß relokatierbar sein. Dies kann bspw. durch ein Kopieren des Boot-Blocks "A" in ein während der Flash-Programmierung freien RAM-Bereich erfolgen. Anschließend wird dann der Boot-Block "A" aus dem RAM ausgeführt. Auch nach einer fehlgeschlagenen Flash-Programmierung des Boot-Blocks muß ein Neustart des Programmierablaufs möglich sein. Zur Erhaltung der Verfügbarkeit nach einem Abbruch ist ein fehlerfreier Boot-Block ausreichend. In einem ersten Schritt des Verfahrens wird der alte Boot-Block "A" in einen freien RAM-Bereich kopiert. In einem zweiten Schritt wird der alte Boot-Block im RAM aktiviert, was durch die Markierung "A" kenntlich gemacht ist, und im Flash-Speicher deaktiviert. Der neue Boot-Block wird in einem Flash-Segment C zwischenabgelegt. Dabei wird das Flash-Segment C zunächst gelöscht, der neue Boot-Block in Flash-Segment C programmiert und eine Signaturprüfung für den neuen Boot-Block im Flash-Segment C durchgeführt. Nach einem Abbruch während dieser Verfahrensschritte kann mit dem gültigen, alten Boot-Block im Flash-Segment A die Flash-Programmierung erneut gestartet werden. In einem dritten Schritt wird der neue Boot-Block programmiert, was durch ein Kopieren von Flash-Segment C nach Flash-Segment A durchgeführt wird. Dieser Schritt umfaßt das Löschen des Flash-Segments A, das Programmieren des neuen Boot-Blocks in Flash-Segment A durch Kopieren des Flash-Segments C nach A und einer Signaturprüfung für den neuen Boot-Block in Flash-Segment

A. Nach einem Abbruch während dieser Verfahrensschritte kann mit dem gültigen, neuen Boot-Block in Flash-Segment C die Flash-Programmierung erneut gestartet werden. Der jeweils gültige Boot-Block im Flash-Speicher muß markiert werden. Diese Gültigkeitsmarkierung selbst muß unverlierbar im Flash-Speicher abgelegt werden, so daß mit dieser Information ein Wiederaufsetzen möglich ist.

Ansprüche

1. Verfahren zur Durchführung eines Software-Updates eines Steuergerätes durch eine Flash-Programmierung eines mehrere Segmente aufweisenden Flash-Speichers des Steuergerätes über eine serielle Schnittstelle, das mindestens die folgenden Schritte aufweist:

a) Festlegen von an die Flash-Programmierung zu stellende Anforderungen;

b) Festlegen eines Ablaufs der Flash-Programmierung durch einen Zustände und Übergänge der Software definierenden Zustandsautomaten;

c) Überprüfen von Verfügbarkeits-, Sicherheits- und Zuverlässigkeitsanforderungen eines jeden Zustands und eines jeden Übergangs des Zustandsautomaten.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für die Software des Steuergerätes verschiedene Betriebszustände, insbesondere ein "Anfangszustand", ein "Normalzustand" und ein "Software-Update", Übergänge zwischen den Betriebszuständen und Übergangsbedingungen spezifiziert werden.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß für Flash-Programmierung relevante Speicherblöcke der Software des Steuergerätes in programmierbare und nicht programmierbare Speicherblöcke unterteilt werden und neu zu programmierende Komponenten der Software den Speicherblöcken entsprechend zugeordnet werden.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß die Speicherblöcke der Software jeweils einem Speicher des Steuergerätes, insbesondere ein programmierbarer Speicherblock mindestens einem Segment des Flash-Speichers, bzw. ein nicht programmierbarer Speicherblock einem ROM des Steuergerätes zugeordnet werden.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß jeweils in Segmenten des Flash-Speichers des Steuergerätes ein für die Durchführung der Flash-Programmierung notwendige Software-Funktionalität bereitstellender Boot-Block, ein Programm- und ein Datenstand und in einem ROM des Steuergerätes ein für die Durchführung der Flash-Programmierung notwendige Software-Funktionalität bereitstellender Start-up-Block abgelegt werden.

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß Sicherheits-, Zuverlässigkeits- und Verfügbarkeitsanforderungen der Flash-Programmierung spezifiziert werden.

7. Verfahren nach einem der Ansprüche 2 bis 6, dadurch gekennzeichnet, daß durch den Zustandsautomaten bei Durchführung der Flash-Programmierung im Betriebszustand "Software-Update" einnehmbare Subzustände, Übergänge

zwischen diesen und Übergangsbedingungen spezifiziert werden.

8. Zustandsautomat zur Durchführung eines Software-Updates eines Steuergerätes durch eine Flash-Programmierung, der alle bei Durchführung des Software-Updates einnehmbare Subzustände der Software des Steuergerätes, Übergänge zwischen diesen und Übergangsbedingungen definiert und bei Auftreten einer Störung während der Durchführung des Software-Updates ein dauerhaftes, unverlierbares Abspeichern eines zuletzt gültigen oder fehlerfrei durchlaufenen Zustandes spezifiziert.

9. Zustandsautomat nach Anspruch 8, dadurch gekennzeichnet daß als Subzustände "Abbruch/Fehlermeldung", "Abschluß/Erfolgsmeldung", Subzustände für eine Authentisierung und Signaturprüfung, Subzustände für Löschen und Programmieren von Segmenten des Flash-Speichers spezifiziert sind.

10. Computerprogramm, bestehend aus Programmcodeelementen durch welches bei Ausführen der Programmcodeelemente auf einem Computer oder auf einem Computersystem automatisch vordefinierte Verfügbarkeits-, Sicherheits- und Zuverlässigkeitsanforderungen eines jeden Zustands und eines jeden Übergangs eines Zustandsautomaten gemäß Anspruch 8 oder 9 überprüft werden.

11. Verfahren zur Durchführung einer Flash-Programmierung eines für die Durchführung der Flash-Programmierung notwendige Software-Funktionalität bereitstellender, in

einem ersten Segment (Flash-Segment A) eines Flash-Speichers abgelegten Boot-Blocks,

wobei das Verfahren mindestens die folgenden Schritte aufweist:

a) Kopieren des neu zu programmierenden, alten Boot-Blocks in einen freien Bereich eines zweiten Speichers (RAM);

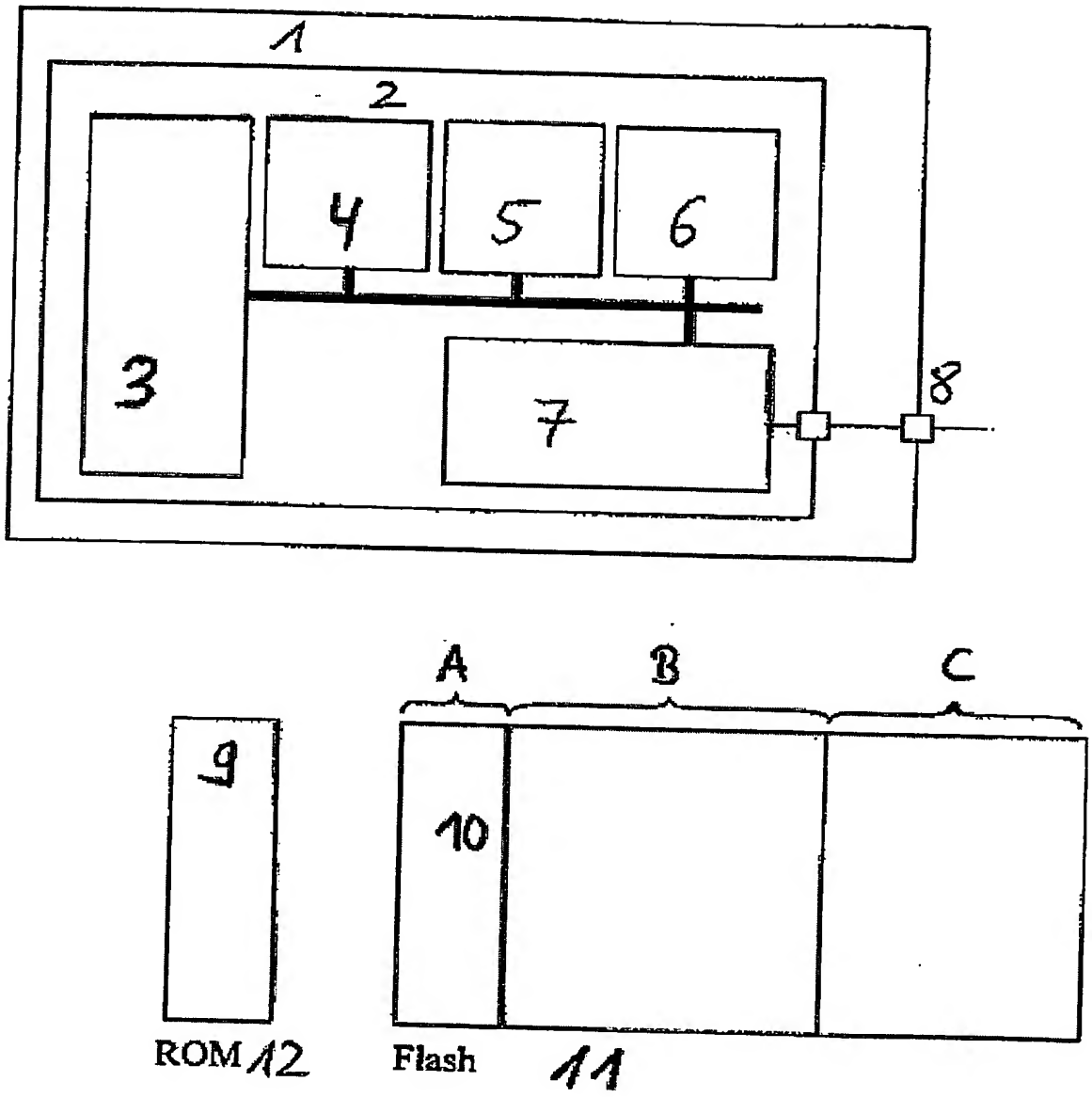
b) Aktivieren des alten Boot-Blocks in dem zweiten Speicher (RAM) und Deaktivieren des alten Boot-Blocks im Flash-Speicher;

c) Zwischenablegen eines neuen Boot-Blocks in einem zweiten Segment (Flash-Segment C) des Flash-Speichers;

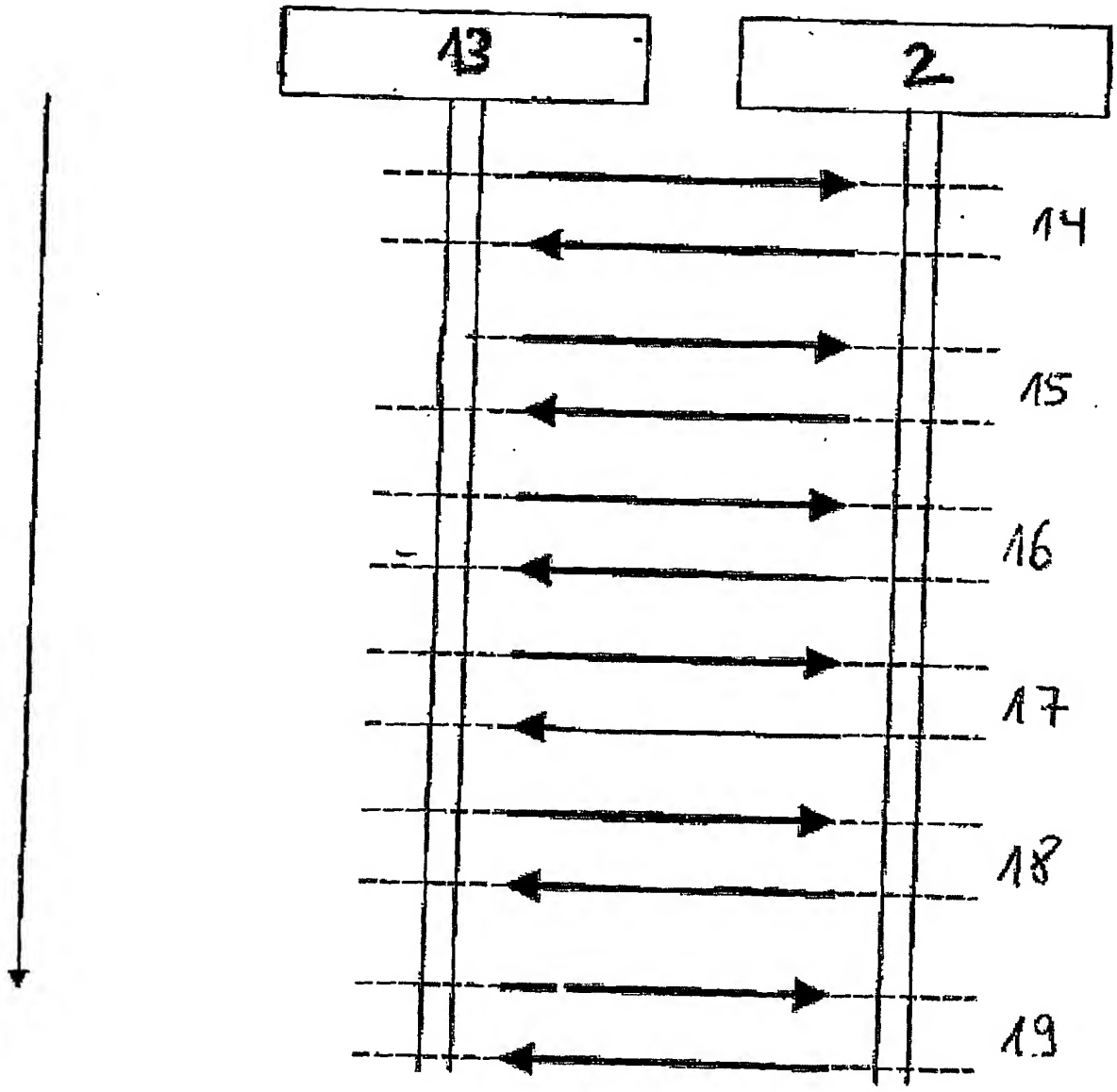
d) Programmieren des neuen Boot-Blocks durch Kopieren des zweiten Segments (Flash-Segment C) nach dem ersten Segment (Flash-Segment A);

e) Aktivieren des neuen Boot-Blocks in dem ersten Segment (Flash-Segment A) und Deaktivieren des alten Boot-Blocks in dem zweiten Speicher (RAM).

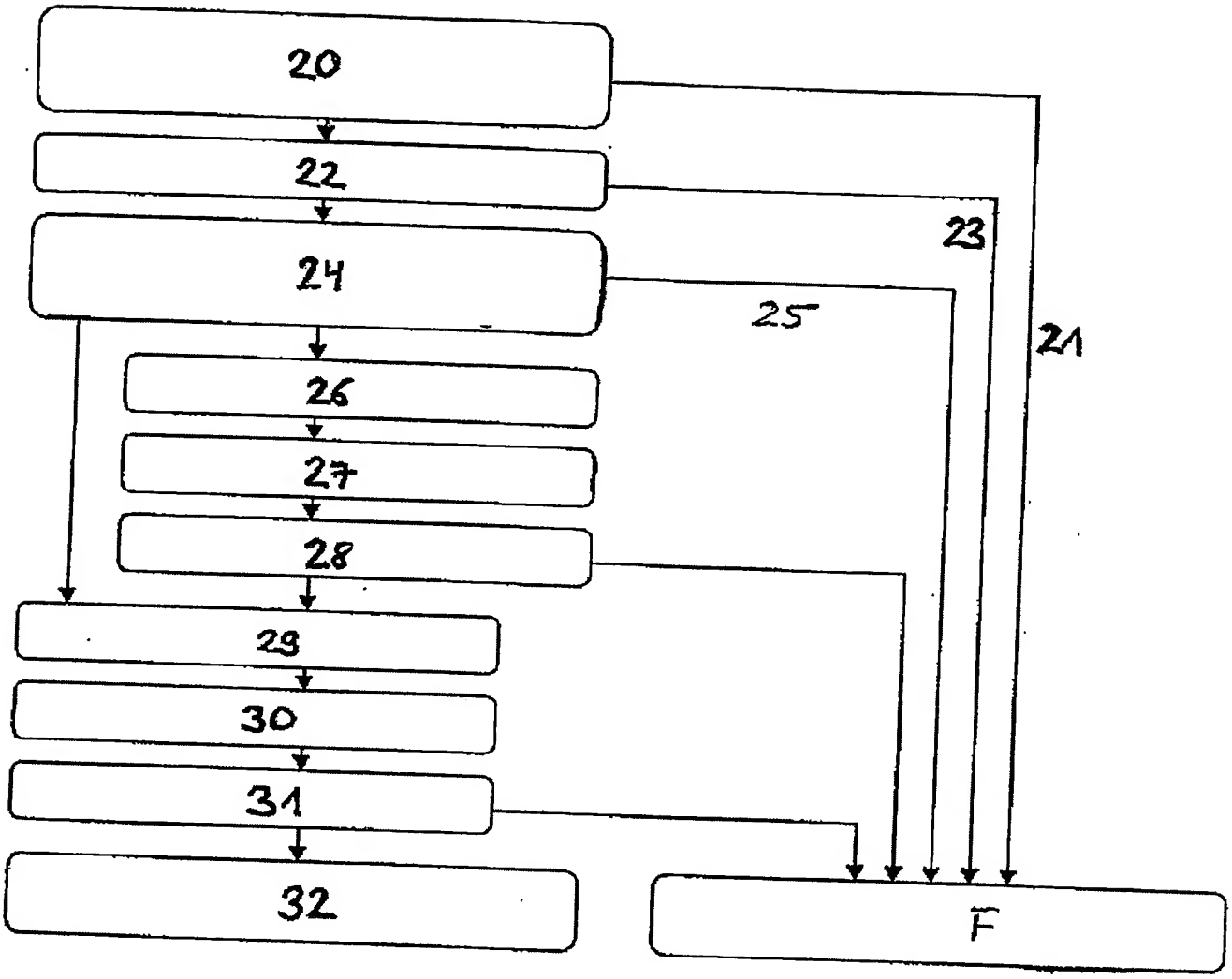
12. Verfahren nach Anspruch 11, dadurch gekennzeichnet, daß im Flash-Speicher immer ein Boot-Block als für einen Neustart der Flash-Programmierung gültiger Boot-Block markiert wird.



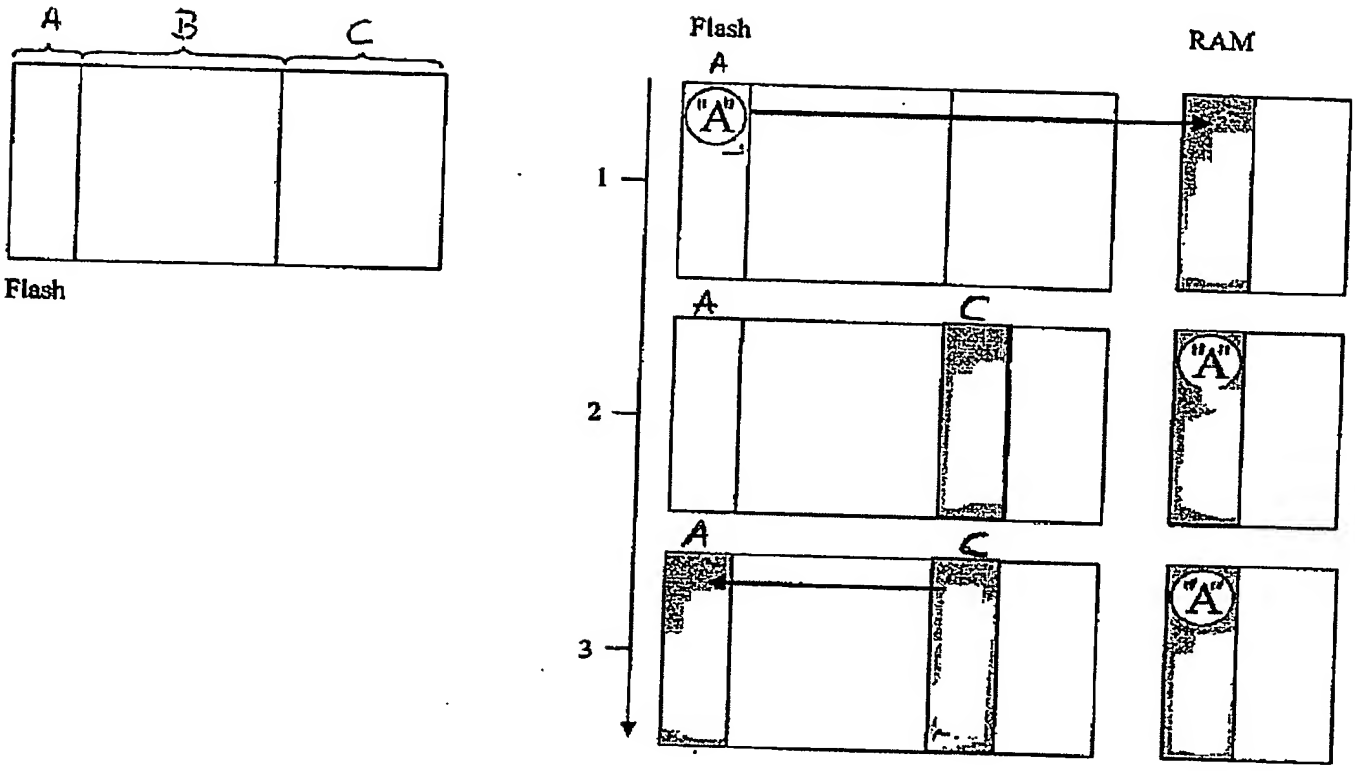
Figur 1



Figur 2



Figur 3



Figur 4